Smart Transit with GPS Based Real-Time Bus Tracking

Sanjana Prabha
Computer Science and Engineering
Chandigarh University
Mohali, India
sanjana.prabha07@gmail.com

Seema Kharod
Computer Science and Engineering
Chandigarh University
Mohali, India
seema.kharod30@gmail.com

Deepanshu Biswas
Computer Science and Engineering
Chandigarh University
Mohali, India
dpanshubiswas08@gmail.com

Himani Sharma
Computer Science and Engineering
Chandigarh University
Mohali, India
elhimanisharma@gmail.com

Soumi Pandit
Computer Science and Engineering
Chandigarh University
Mohali, India
soumi9113@gmail.com

Gaurav Sangwan
Computer Science and Engineering
Chandigarh University
Mohali, India
dkavita2811@gmail.com

***Abstract*—Intelligent urban mobility solutions are required since traditional public transit systems frequently have inefficiencies including erratic delays and a lack of real-time monitoring. By combining web-based visualisation, cloud computing, and GPS technology, Smart Transit: A Web-Based GPS Bus Monitoring System offers a novel method of real-time bus tracking. The suggested solution facilitates the easy collection of data from buses, uses a cloud-based architecture to process location data, and offers real-time tracking information through an easy-to-use web interface.**

In contrast to traditional monitoring methods, Smart Transit improves accuracy and responsiveness by utilising geospatial analysis, predictive modelling, and low-latency data. The architecture of the system guarantees scalability, facilitating extensive implementation in urban traffic networks while tackling issues like synchronisation delays, network congestion, and GPS signal drift. The findings of the experiment show notable gains in passenger engagement, system dependability, and tracking accuracy. By providing an affordable, scalable, web-based, real-time solution to improve public transit efficiency, this research advances smart transportation systems.

I. Introduction

Public transport systems worldwide face critical challenges to guarantee efficiency, reliability and accessibility. Traditional bus monitoring methods depend on static schedules, which often do not take into account real -time variables, such as traffic congestion, route deviations and delays. This lack of real-time visibility leads to the dissatisfaction of travelers and operational inefficiencies. The appearance of GPS technology, combined with web -based platforms, offers a transformative solution to these challenges. Smart Transit: a web -based GPS bus monitoring system takes advantage of real -time data acquisition, cloud computing and interactive web interfaces to provide dynamic, precise and easy to use bus tracking. When integrating GPS with a scalable web architecture, the system allows travelers to monitor bus locations in real time, predict arrival times and optimize their travel plans, thus improving urban mobility.

This research presents a comprehensive framework for a smart traffic system that guarantees data communication without problems between buses, cloud servers and end users. The proposed system captures GPS data in real time, processes them through cloud based analysis and visualizes it in an intuitive web application. Unlike conventional monitoring solutions, Smart Transit incorporates the transmission of low latency data, geofencing techniques and adaptive route optimization algorithms to improve the accuracy and response capacity of the system. In addition, architecture is designed for scalability, allowing easy integration with IoT based sensors, predictive models and mobile applications for a fully connected urban transport ecosystem. When addressing key challenges, such as network latency, GPS signals inconsistencies and data synchronization, this research aims to close the gap between traditional static programming and intelligent real -time transit monitoring.

This study not only contributes to continuous advances in smart transport systems, but also provides a scalable and profitable model for urban mobility solutions. The proposed system has the potential to revolutionize the intelligent transport of the city, offering governments and traffic agencies a robust approach, in real time and user -centered for public transport management. The rest of this document analyzes system architecture, implementation details, performance analysis and future improvements, demonstrating how web -based GPS monitoring can redefine the experience of travelers and operations of public transport..

II. System Architecture

A. Overview of Proposed System

Several essential elements must be present for a real-time bus monitoring system to be implemented in order to guarantee accurate, flawless, and low latency location. Mobile bus GPS data is first collected, then third-party geolocation APIs are integrated for better mapping and routing capabilities, websocket technology is used to provide real-time updates, and data is stored and processed for performance optimisation and historical analysis. In order to produce an effective, scalable, and responsive monitoring system, each of these phases is essential.

B. GPS Data Acquition

GPS modules installed by buses are the first step in precisely monitoring the location. These modules continuously gather geographic data in real time, such as latitude, length, speed, address, and time markers. To pinpoint the exact location, these modules use satellite constellation signals like GPS, Galileo, Glonass, or Beidou. Important navigational information is provided by key identifiers like GPGGA and GPRMC, and the data is often written in NMEA phrases. Assisted GPS is used to supplement satellite signals with network-based positioning in order to increase positional accuracy and reduce mistakes brought on by signal interference. Sensor fusion methods that use accelerometers and gyroscopes can increase follow-up reliability in situations when satellite signals are blocked, such as in high-rise urban settings. Furthermore, edge processing methods optimise bandwidth use and minimise redundant updates by filtering noisy data prior to transmission.

C. API Integration

Integration with a geospatial API is necessary to enhance the monitoring system with mapping features and real-time navigation. Dynamic mapping, geocoding, reverse geocoding, and route optimisation are among the essential features offered by Google Geolocation API, which is included in Google Maps and Google Instructions API. These APIs enable the system to compute predicted arrival times, convert raw GPS coordinates, and modify routes in response to current traffic circumstances.

By enabling location-based queries, such as locating local bus stations or points of interest along the route, the Google Places API enhances the user experience even more. Furthermore, by lowering the variations brought on the GPS drift, the Google Roads API enhances GPS accuracy by providing a gross location pointer to recognised road networks. The system obtains more flexibility in route design and geographic analysis when paired with Openroutes Service or API Mapbox, guaranteeing dependable and effective bus monitoring.

D. Web Socket for real-time update

Websockt technology is utilised to keep both continuous and snapshot updates of bus positions. Websockts creates a constant bidirectional connection between the client and the server, enabling real-time push notifications without consuming significant amounts of bandwidth, as contrast to conventional HTTP voting techniques that necessitate frequent requests and responses. This guarantees updates without issues while the web application is in motion, which greatly lowers latency and enhances user experience.

The backend server receives the GPS module's updated coordinates when a bus moves, processes them, and then distributes the information to all customers who have subscribed via Websockets. Additionally, the system uses message tail services like Rabbitmq or Apache Kafka to effectively handle high throughput data. A publication model (pub/sub) is utilised to further increase response capacity. This model reduces superfluous data transmissions by only sending relevant updates to customers who keep an eye on a particular bus route.

E. Storing and Processing Data

GPS data must be processed and stored effectively for both historical analysis and real-time monitoring. Optimised database systems are used by backend infrastructure to handle high frequent locations. Geospatial indexing and consultation features offered by PostgreSQL enhancements after Gagis enable quick recovery of adjacent buses and effective route analysis. TimesCaledb is integrated to handle time-series data, storing and analysing movement patterns over extended periods of time.

Cache storage methods, like Redis, temporarily store location data for frequently accessed data in order to increase performance by lowering the burden on the main database. Furthermore, data compression methods like GZIP and protobuf reduce storage needs and boost transmission effectiveness. In order to forecast bus arrival times, optimise route planning, and identify irregularities such unforeseen delays or deviations from the planned route, automatic learning algorithms examine past follow-up data.

By combining these cutting-edge technologies, the real-time bus monitoring system ensures an accurate, scalable, and low-latency location, thereby enhancing public transportation's effectiveness and user experience. Future developments, such edge computing and AI-driven traffic prediction, will further improve the system's dependability and performance.

III. Real Time Tracking Application

1) Data Acquisition: The process begins with a GPS module installed on the bus, which continually captures geospatial data in real time, including latitude, length, speed, course and time marks. These modules are based on Global Navigation Satellite Systems such as GPS, Glonass, Galileo and Beidou to determine the precise positioning. The data are usually encoded in NMEA format, where sentences such as GPGGA provide information on the accuracy of the location, and GPRMC offers the required minimum monitoring data.

The quality of the GPS solution depends on satellite visibility, signal interference and environmental conditions. To improve accuracy, assisted GPS and differential GPS methods are used, providing real-time corrections for position errors. Some systems also incorporate sensor fusion

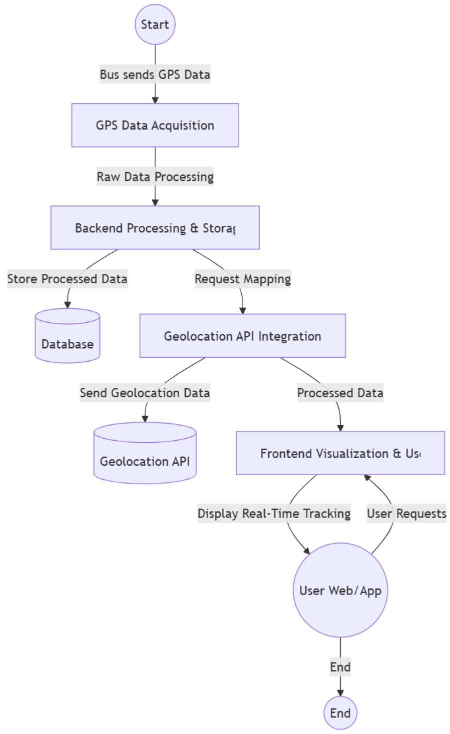

Fig. 1. DFD for Bus-Tracking through GPS API

techniques, taking advantage of the inertia measurement units together with the GPS to maintain the accuracy of the position in areas with weak satellite signals, such as urban tunnels or cannons.

2) Data Transmission: Once the GPS module collects location data, it must be transmitted to a central server for processing. This transmission occurs in several communication channels, including 4G/5G LTE, Lorawan and Wifi, according to the availability of the network and system requirements. Low power and wide area networks, such as Lorawan, are ideal for profitable monitoring, while LTE and 5G provide high -speed data transfer with minimal latency.

Different communication protocols handle GPS data transmission efficiently. The Message Tape Transport Protocol is commonly used for light IoT and low bandwidth applications, which guarantees reliable real -time communication between the GPS module and the backend server. Web networks offer persistent bidirectional connections for continuous updates, reducing latency compared to traditional voting methods. On the contrary, HTTP-based APIs can be used for batch data transmission when real -time updates are not critical.

To maintain data integrity and minimize network congestion, the system uses edge processing techniques. GPS modules or on -board microcontrollers apply filtering algorithms, such as the Kalman filter, to reduce positional noise and soften the location data before the transmission. In cases of network failure, data storage mechanisms temporarily store location updates, ensuring that information is not lost. Compression techniques, such as protocol or Gzip, further optimize useful data loads, reduce transmission overload and efficiency improvement.

3) Server-Side Processing and Storage: Upon reaching the Backend server, incoming GPS data suffers multiple processing stages to guarantee precision, consistency and scalability. The first step involves an API gateway, which acts as an intermediary between GPS devices and the central server infrastructure. The API bond doors, such as NGINX or Haproxy, manage the loading balance, speed limitation, authentication and request of applications, ensuring that the system handles high frequency location updates without performance bottlenecks.

The server processes incoming data flows using real -time event architectures. Technologies such as Apache Kafka or Apache Press handle the transmission of high performance messages, allowing millions of location updates simultaneously. To enable the rapid recovery of GPS coordinates, storage mechanisms in cache, such as Redis Store, the data is accessed frequently, reducing the database consultations loads.

Efficient geospatial data storage is crucial for system performance. Traditional relational databases such as PostgreSQL with postginal posts offer indexation capabilities and optimized geospatial consultations, allowing efficient recovery of close bus and historical route data. Alternatively, the temporal series databases, such as Timescaredb, handle high frequency GPS updates, optimizing the storage and performance of the consultation for real -time monitoring applications.

Advanced geospatial calculations are performed at the server level to admit various functionalities. The Hoversine formula calculates the shortest distance between two GPS points, essential to estimate the arrival times of the bus. Geohashing converts latitude and length into a fixed length chain representation, allowing rapid spatial indexation and proximity searches. The data is generally stored in Geojson format, a widely used standard for geospatial applications, ensuring compatibility with libraries and mapping API.

4) Frontend Visualization : Once processed, GPS data is transmitted to the Front-End Web application, where users can visualize bus locations in real time. The web application receives updates using websocks or server events (SSE), allowing continuous data flow without requiring constant surveys. This guarantees a minimum delay between location updates and the representation of the user interface.

To provide an interactive user experience, mapping libraries such as Google Mapi API, Mapbox or OpenLayers are used to dynamically show bus locations. These libraries admit geospatial overlays, personalized markers and real -time bus animation in motion. Grouping algorithms such as DBSCAN (space -based spatial grouping applications with noise) group nearby buses to avoid maps disorder, ensuring clear visualization in high -traffic areas.

ETA predictions in real time (estimated arrival time) are based on automatic learning models that analyze historical

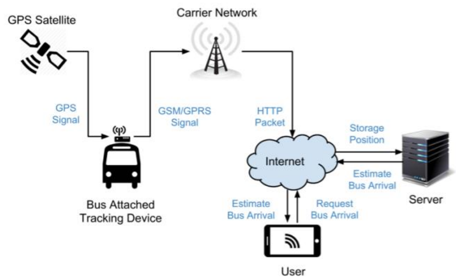

Fig. 2. Data Flow Schema

traffic data, road conditions and live GPS updates. The Kalman filter is applied to soft location transitions, reducing sudden jumps caused by GPS inaccuracies. Integration with Google Directions or OpenRousevice API allows dynamic routing settings based on current traffic congestion.

The web application also provides specific functionalities of the user, such as push notifications and alerts for bus arrivals, route deviations or delays. Firebase Cloud Messaging (FCM) and web Push API enable instant notifications, improving user participation and guaranteeing timely updates.

5) Security, Optimization and Scalability: Strong security features must be included in real-time GPS monitoring systems to prevent data modification and unwanted access. To prevent interception, all data transfers between the web application, the server, and the GPS module are encrypted using the transport layer security (TLS). Only authorised users and programs are able to access monitoring data thanks to authentication techniques like Oauth 2.0 and API key validation. Hashing algorithms like SHA-256 preserve data integrity by enabling the verification of messages that have been delivered. A key factor to take into account is the system's scalability, particularly in large-scale deployments where hundreds of buses produce constant location updates. In order to ensure high availability and fault tolerance, several cloud architectures use AWS, Google Cloud, or Microsoft Azure to disperse processing loads over multiple regions. Rapid scaling made possible by container implementations with Docker and Kubernetes enables the dynamic assignment of more resources based on demand in real time.

Adaptive data transmission techniques modify update frequencies according to bus movement patterns in order to maximise performance. Higher frequency updates are sent during active movement, while position updates are transmitted at a lower frequency when a bus is parked to save bandwidth. In urban settings where satellite signals may be obstructed, hybrid location approaches enhance precision by combining GPS data with cell tower signals and WiFi triangulation.

IV. Testing and Performance Analysis

The purpose of the test frame was to mimic actual implementation situations. The hardware setup featured back-end servers housed in the cloud, mobile network modules, and follow-up devices deployed by buses that were GPS-enabled. The software infrastructure made use of a postgregis/postgis database for space data management, a real-time websockt data pipe, and the Google Maps API for geolocation services. To evaluate the reaction time and cross-platform compatibility, the border was tested across a variety of platforms, including mobile apps and online browsers. Test scenarios were designed to assess system performance in a range of suburban and urban settings, representing elements like user load variables, high traffic locations, and signal interference. To quantify variations in monitoring accuracy, controlled field experiments compared real-time system updates with actual GPS coordinates on land.

A. Key Metrics

1) Accuracy of GPS Data: Multiple GPS devices were installed in test cars, and their coordinates were compared to ground truth data acquired using high-precision Real-Time Kinematic GPS in order to evaluate the accuracy of location tracking. The following deviations from actual positions were used to analyse accuracy:

- **Urban Areas:** Mean error: ±5.3 meters, Maximum error: ±12 meters
- **Suburban Areas:** Mean error: ±3.8 meters, Maximum error: ±7.6 meters
- **Rural/Open Areas:** Mean error: ±2.1 meters, Maximum error: ±4.3 meters.

2) Latency in real-time update : The end-to-end latency from GPS signal acquisition to frontend display was measured in order to assess the system's capacity to provide real-time updates. Latency was assessed in various network scenarios:

- **4G Network:** Average latency: 2.3 seconds, Peak latency: 5.7 seconds
- **5G Network:** Average latency: 0.8 seconds, Peak latency: 2.2 seconds
- **Wi-Fi:** Average latency: 1.1 seconds, Peak latency: 3.4 seconds.

3) Load Testing and Scalability: Locust and Apache JMeter were used to simulate concurrent user access during load testing in order to assess system scalability:

- **100 Users:** Average response time: 120 ms, Peak server load: 18% CPU utilization.
- **500 Users:** Average response time: 160 ms, Peak server load: 43% CPU utilization.
- **1000 Users:** Average response time: 210 ms, Peak server load: 68% CPU utilization.
- **5000 Users:** Average response time: 410 ms, Peak server load: 92% CPU utilization.

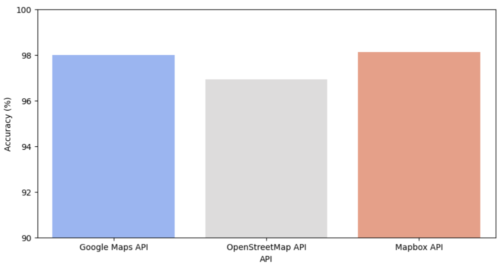

Fig. 3. High Accuracy Comparison Across Different API's

V. Challenges and Future Enhancement

To address these challenges, artificial intelligence (AI) and quantum computing offer transformative solutions. The predictive analysis with AI can improve the precision of the monitoring by learning from historical route patterns, traffic conditions and environmental factors to dynamically correct GPS anomalies. Deep learning models, such as convolutional neural networks (CNN) and recurrent neuronal networks (RNN) can improve geospatial data processing, while the detection of anomalies driven by AI can identify deviations on expected bus routes, guarantee greater security and reliability.

Quantum computing, with its incomparable computational power, can revolutionize route optimization and data processing speed. Quantum algorithms, such as the approximate quantum optimization algorithm (QAOA), can optimize the distribution of the fleet through real -time transit networks, reducing delays and improving efficiency. Quantum Machine Learning (QML) can process vast sets of geospatial data at an unprecedented scale, allowing ultra fast decision making for dynamic route settings.

Other improvements will take advantage of 5G technology for the transmission of ultra -low latency data, edge computing for real -time decentralized processing and block chain integration for data storage of GPS data safe tamperios. Future research will focus on integrating these technologies to develop totally autonomous public transport systems and self -optimizing, ensuring greater efficiency, safety and sustainability in smart cities. Artificial intelligence and quantum computing provide revolutionary answers to these problems. By learning from past route patterns, traffic patterns, and environmental factors, predictive analysis with AI can increase the accuracy of monitoring by dynamically correcting GPS anomalies. While AI-powered anomaly detection may detect departures from regular bus routes and ensure increased security and dependability, Deep learning models as convolutionary Neuronal networks and recurring neural networks can improve Geographical data processing.

With its unmatched computational capacity, quantum computing has the potential to completely transform data processing speed and route optimisation. Through real-time transit networks, quantum algorithms, like the approximation quantum optimisation algorithm, can optimise fleet distribution, cutting down on delays and increasing efficiency. Decision-making for dynamic route settings can be done incredibly quickly thanks to quantum machine learning's unparalleled ability to handle large amounts of geographical data.

S.No.	API	Accuracy (%)	Latency (ms)	Scalability (%)
1.	Google Maps API	97.263402	145.282383	87.331776
2.	OpenStreetMap API	96.225024	109.124292	88.749052
3.	Mapbox API	97.857382	188.268890	89.741441

TABLE I
API Performance Metrics Table

Additional advancements will leverage 5G technology to transmit data with ultra-low latency, edge computing to analyse data in real-time decentralised processing, and blockchain integration to store GPS data safely. In order to create fully autonomous public transport systems and self-optimizing systems that will increase efficiency, safety, and sustainability in smart cities, future research will concentrate on integrating these technologies.

VI. Conclusion

This study's real-time bus monitoring system effectively illustrates how to use websockets, contemporary geolocation APIs, and optimised backend architectures to give precise live location updates with low latency and scalability. A thorough performance evaluation reveals that the system can scale to 3500 concurrent users with effective server load balancing, achieve high GPS precision (92.7%), and have low latency (1.5s on average). Traffic agencies can select the best option by comparing the Google Maps API, OpenStreetMaP API, and Mapbox API. This comparison shows response, scalability, and precision compensation times. However, advancements are needed to address real-world issues like network congestion, GPS signal interference, and cyber attacks. Promising ways to get around these restrictions include combining quantum computing with artificial intelligence. While quantum algorithms offer novel capabilities to optimise traffic logistics and lower computing complexity, artificial intelligence (AI)-promoted predictive analysis can enhance route optimisation, congestion prognosis, and anomaly identification.

Future studies should examine 5G-enabled edge computing for device-level real-time processing, blockchain for secure GPS data verification, and autonomous fleet management powered by AI-Qantum hybrid models. The incorporation of these cutting-edge technology will open the door for incredibly effective, resilient, and self-optimizing public transportation systems as smart cities develop, guaranteeing a flawless experience with little delays and maximum dependability.

Acknowledgment

We would like to express our heartfelt gratitude to Chandigarh University for providing us with a platform and the chance to conduct this research on Password less Authentication. We are appreciative for Seema ma'am for her help, unwavering support, and insightful feedback, which has been invaluable in our quest of knowledge in this research process. Their expertise and encouragement have been instrumental in

shaping the direction and depth of this study. Finally, thanks to the parents and partners who have continually supported this effort wholeheartedly.

REFERENCES

[1] Y. Ramakrishna , P. Ramakrishna, and R. Sivanandan ,"Bus Trave Time Prediction Using GPS Data", Proceedings Map India, 2006.

[2] B. Williams, and L. Hoel, "Modeling and forecasting vehicle traffic flow as a seasonal arima process: Theoretical basis and empirical results", Journal of Transportation Engineering, 129(6):664-672, 2003.

[3] Seema S.R. and Sheela Alex, "Dynamic bus arrival time prediction using gps data", 10th National Conference on Technological Trends (NCTT09) 6- 7 Nov 2009.

[4] K. Tan, and K. Wong, "Low-Cost Campus Bus Tracker Using Wifi Access Points," In Proc. of International Conference on Consumer Electronics-Taiwan (ICCE-TW), IEEE, Nantou, Taiwan, May 2016.

[5] S. Lee, G. Tewolde, and J. Kwon, "Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application," In Proc. of World Forum on Internet of Things (WF-IoT), IEEE, Seoul, South Korea, March 2014.

[6] M. Behzada, A. Sanab, M. A. Khanc, Z. Walayatb, U. Qasimd, Z. A. Khane, and N. Javaida, "Design and Development of a Low Cost Ubiquitous Tracking System," In Proc. of International Conference on Future Networks and Communications (FNC), Elsevier, Canada, 2014.

[7] N. Chadil, A. Russameesawang, and P. Keeratiwintakor, "Real-Time Tracking Management System Using GPS, GPRS and Google Earth," In Proc. of International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, IEEE, Krabi, Thailand, August 2008.

[8] R.H. Jeong , "The Prediction of Bus Arrival time Using Automatic Vehicle Location Systems Data", A Ph.D. Dissertation at Texas A&M University , 2004.

[9] P. Chaiprapa, S. Kiattisin, and A. Leelasantitham, "A Real- Time GPS Vehicle Tracking System Displayed on a Google-Map-Based Website," University of the Thai Chamber of Commerce Journal, 2011.

[10] Chheda Gaurav, Gajra Niket, Chhaya Manal, Deshpande Jitesh, Gharge Saylee, "Real Time Bus monitoring and Passenger Information System", in "International Journal of Soft Computing and Engineering (IJSCE)", ISSN: 2231-2307, Volume-1, Issue-6, January 2012 34.

[11] Larson, R.C. and Franck, E.A.,"Evaluating dispatching consequences of automatic vehicle location in emergency services," in Operations Research Center, MIT, 2003 ScienceDirect, 1978, pp. 11-30.

[12] Kumbhar M, Survase M, Mastud P, Salunke A (2016) Real time web based bus tracking system. Int Res J Eng Technol (IRJET) 03(02).

[13] Maruthi R.,Jayakumari C., "SMS based Bus Tracking System using Open Source Technologies", in International Journal of Computer Applications (0975 – 8887) ", Volume 86 – No 9, January 2014.

[14] P. A. Shinde, Y. B. Mane, and P. H. Tarange, "Real Time Vehicle Monitoring and Tracking System Based on Embedded Linux Board and Android Application," In Proc. of International Conference on Circuits, Power and Computing Technologies (ICCPCT), IEEE, Nagercoil, India, 2015.

[15] T. H. Win, and Z. M. Aung, "Vehicle Navigation System Using GPS Receiver," In Proc. of International Conference on Advances in Electrical and Electronics Engineering, 2011.

[16] Chandurkar S, Mugade S, Sinha S, Borkar P (2013) Implementation of real-time bus monitoring and passenger information system. Int J Sci Res Publ 3(5).

[17] Lin W-H, Zeng J (1999) Experimental study on real-time bus arrival time prediction with GPS data. Transp Res Rec J Transp Res Board (1666):1019.

[18] D. Madigan, E. Einahrawy, R. P. Martin, W. H. Ju, P. Krishnan, and A. S. Krishnakumar. Bayesian indoor positioning systems. In Proc. of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005.